Searching for Linguistic Phenomena in Literary Digital Libraries*

Felipe Sánchez-Martínez, Mikel L. Forcada, and Rafael C. Carrasco

Departament de Llenguatges i Sistemes Informàtics
Universitat d'Alacant, E-03071 Alacant, Spain
{fsanchez,mlf,carrasco}@dlsi.ua.es

Abstract. This paper describes a set of tools and Java classes that allow the Lucene text search engine to use morphological information to index and search; in particular, it describes the use of the linguistic resources developed for the Apertium open-source machine translation platform to extract morphological information while indexing. We describe which linguistic information is automatically obtained, how to use it when indexing new documents with Lucene, and how linguistic attributes can be used to specify query terms. The use of morphological information makes it possible to search for specific linguistic phenomena, and to explore in a richer way the cultural heritage in current digital libraries.

1 Introduction

We describe a set of tools and Java classes that have been implemented to allow the Lucene text search engine[1] to use morphological information while indexing and searching. Morphological information, such as part-of-speech (PoS) and inflection information, enriches the indexes and makes Lucene capable of processing smarter queries in which morphological attributes can be used to specify query terms. This allows searching for specific linguistic phenomena, and exploring in a richer way the cultural heritage in current digital libraries.

To test this approach we have used the morphological dictionaries and PoS taggers developed for the open-source machine translation platform Apertium [1], which has several languages available. This set of tools has been released as open source and can be freely downloaded from `http://sf.net/projects/apertium`, package name `apertium-morph`.

Biemann et al. [2] describe a similar approach in which information about the PoS is included if a PoS tagger is available. We also use PoS information, but in our case we use it even when no PoS tagger is available, since the Apertium dictionaries provide all of the possible PoS tags for each word and, therefore, they can be included in the index at the cost of losing some precision. Resnik and Elkiss [3] describe a linguistic search engine that uses a *query by example*

* Work supported by the Spanish Ministry of Education and Science through project TIN2006-15071-C03-01.

[1] `http://lucene.apache.org`

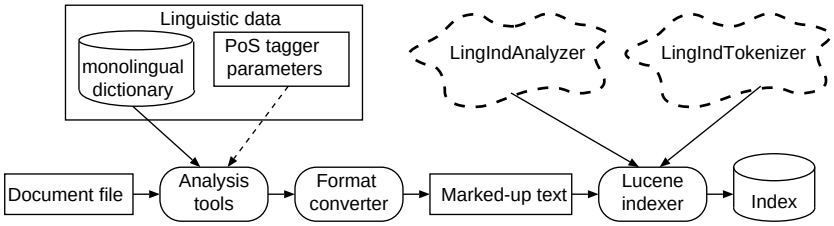

Fig. 1: Scheme of the process of indexing a new document by using linguistic information (see section 3); the dotted arrow means optional data.

procedure and builds a parse tree from the query example before searching; in contrast our approach uses Lucene's standard query language, this means that no parsing or PoS tagging is performed before searching.

The next section briefly overviews the open-source machine translation platform Apertium; then, Section 3 explains how to index new documents with Lucene through this new framework; Section 4 illustrates and evaluates the use of linguistic attributes to search for two specific linguistic phenomena in Spanish. The paper ends with some concluding remarks.

2 Overview of Apertium

Apertium[2] [1] is an open-source machine translation platform that follows a shallow-transfer approach [4, p. 75]; its engine is completely independent from the linguistic data used to translate between a particular pair of languages. Linguistic data is coded using XML-based formats; this allows for interoperability, and for easy data transformation and maintenance. In particular, files encoding linguistic data can be used for purposes other than machine translation.

To test the approach we describe in this paper we have only used the monolingual (morphological) dictionaries and the parameters of the PoS taggers provided by each linguistic package; note, however, that the use of the PoS taggers is optional, as will be shown below. The *monolingual* dictionary of a given language provides for each word (*surface form*) in that language its possible *lexical forms*, each one consisting of lemma, PoS and morphological inflection information; for those words with more than one lexical form, the PoS tagger chooses one of them according to the lexical forms of neighboring words.

3 Indexing new documents

The following steps are followed to index a new document (see figure 1): (i) the text to be indexed is separated from the format information (RTF, HTML, OpenDocument, etc.) and analyzed using a morphological analyzer and a PoS

[2] `http://www.apertium.org`

tagger (if available) for the document language; (ii) the analyzed text is converted into the marked-up format expected by the Java classes used to index the document; and (iii) the document is indexed by using a Lucene-compliant analyzer and tokenizer we have developed so as to properly interpret the marked-up text.

After indexing, the following information is available for each word: its *surface form*, its lemma, PoS and selected morphological information such as the tense in the case of a verb.[3] For instance, the information available for the words in the sentence *Blair does not resign* is:

Word	Surface form	Lemma	Morph. information
Blair	`blair`	`blair`	`noun.person`
does	`does`	`do`	`verb.present`
not	`not`	`no`	`adverb`
resign	`resign`	`resign`	`verb.infinitive`

4 Searching for documents

Queries are written in the language accepted by Lucene's query parser; in the query special prefixes differentiate traditional and morphological terms: the prefix `lem#` is used for lemmas (e.g., `lem#do`) and the prefix `tag#` is used for PoS tags and morphological information (e.g., `tag#verb.infinitive`). After searching, matched words are highlighted in the retrieved documents.

To illustrate this approach we have compared the results obtained for two queries, made over texts of the Spanish news agency EFE[4], when a PoS tagger is used to build the index with those achieved when no PoS tagger is used (i.e., all possible tags of each word are indexed):

`"lem#dignar tag#prep"` searches for any form of the Spanish verb *dignar* followed by a preposition.

`"lem#deber de tag#verb.infinitive"` searches for any form of the Spanish verb *deber* followed by preposition *de* and by a verb in infinitive tense. Note that this query does not disambiguate between the Spanish verb *deber* and the Spanish noun *deber*, as both share the same lemma (therefore, it is only an approximate way to search for this specific linguistic phenomenon).

Table 1 shows the results achieved by the two queries when searching both in the index built by using a PoS tagger and in the index built without PoS tagger; an example of the output produced by each query is also given. Note the large difference between the results achieved by the first query and the second one. The first query achieves better results when a PoS tagger is used to build the index; however, the second query achieves the same results in both cases. A possible explanation of this large difference is that longer queries virtually

[3] When no PoS tagger is used, ambiguous words are indexed by including in the index all of their possible disambiguations at the same position.

[4] `http://www.efe.com`

	Index	Precision	Recall	F-measure
`"lem#dignar tag#prep"`	with PoS tagger	60%	100%	75%
	without PoS tagger	2%	100%	5%
`"lem#deber de tag#verb.infinitive"`	with PoS tagger	82%	100%	90%
	without PoS tagger	82%	100%	90%

`"lem#dignar tag#prep"`	... se han **dignado a** recibirnos ... ... una sede **digna de** mención ... *
`"lem#deber de tag#verb.infinitive"`	... la ONU **debe de reaccionar** urgentemente ... ... no **debiera de estar** representado ...

Table 1: Results (top) achieved by each query, both when searching in the index built by using a PoS tagger and when searching in the index built without PoS tagger, and example of the output produced by each query (bottom). The star marks incorrect retrievals; note that *digna* can be an adjective (lemma *digno)* or a form of the verb *dignar*, and *debe* can be a noun (lemma *debe*) or a form of the verb *deber*.

disambiguate the text at search time, which avoids matching sequences of words that a PoS tagger would never disambiguate in that way.

5 Concluding remarks

This paper has shown how to use linguistic data in order to extract more information about the words appearing in the documents to index. It has also shown the type of queries that can be made by using morphological information to define query terms, and the results achieved when a PoS tagger is used to disambiguate the text to index and when no PoS tagger is used.

The use of morphological attributes to define query terms makes it possible to search for specific linguistic phenomena; this may ease the access and study of the cultural heritage found in current digital libraries. In the near future we plan to integrate this tool with the *Biblioteca Virtual Miguel de Cervantes*.[5]

References

1. Armentano-Oller, C., Carrasco, R.C., Corbí-Bellot, A.M., Forcada, M.L., Ginestí-Rosell, M., Ortiz-Rojas, S., Pérez-Ortiz, J.A., Ramírez-Sánchez, G., Sánchez-Martínez, F., Scalco, M.A.: Open-source Portuguese-Spanish machine translation. In: Computational Processing of the Portuguese Language. Volume 3960 of LNAI. Springer-Verlag (2006) 50–59
2. Biemann, C., Quasthoff, U., Wolff, C.: Linguistic corpus search. In: Proceedings of 4th International Conference on Language Resources and Evaluation. (2004)
3. Resnik, P., Elkiss, A.: The linguist's search engine: an overview. In: Proceedings of the ACL 2005 on Interactive poster and demonstration sessions. (2005) 33–36
4. Hutchins, W.J., Somers, H.L.: An Introduction to Machine Translation. Academic Press (1992)

[5] `http://www.cervantesvirtual.com`